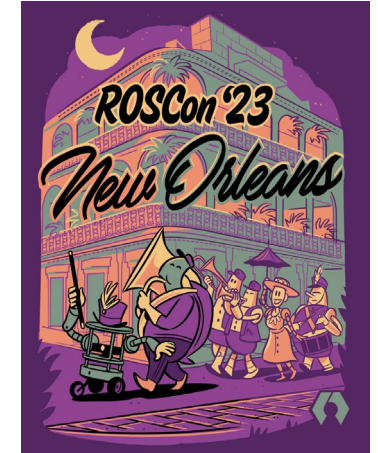


PostgreSQL / PostGIS to ROS2 Bridge for Spatial Data

ROSCon 2023 – New Orleans



Marco Wallner

Markus Hofstätter

Matthias Schörghuber



GitHub



CONTENT

01

MOTIVATION

Spatial Data and Robotics

02

POSTGIS TO ROS2 BRIDGE

Feature Overview

03

EXAMPLES

Gravel Quarry and Real World Use Cases

04

CONFIGURATION

Setup and Examples

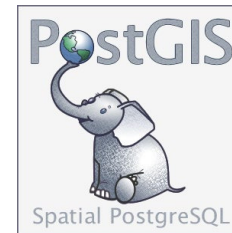
05

EXTENSIBILITY

Adding a Custom Message Type

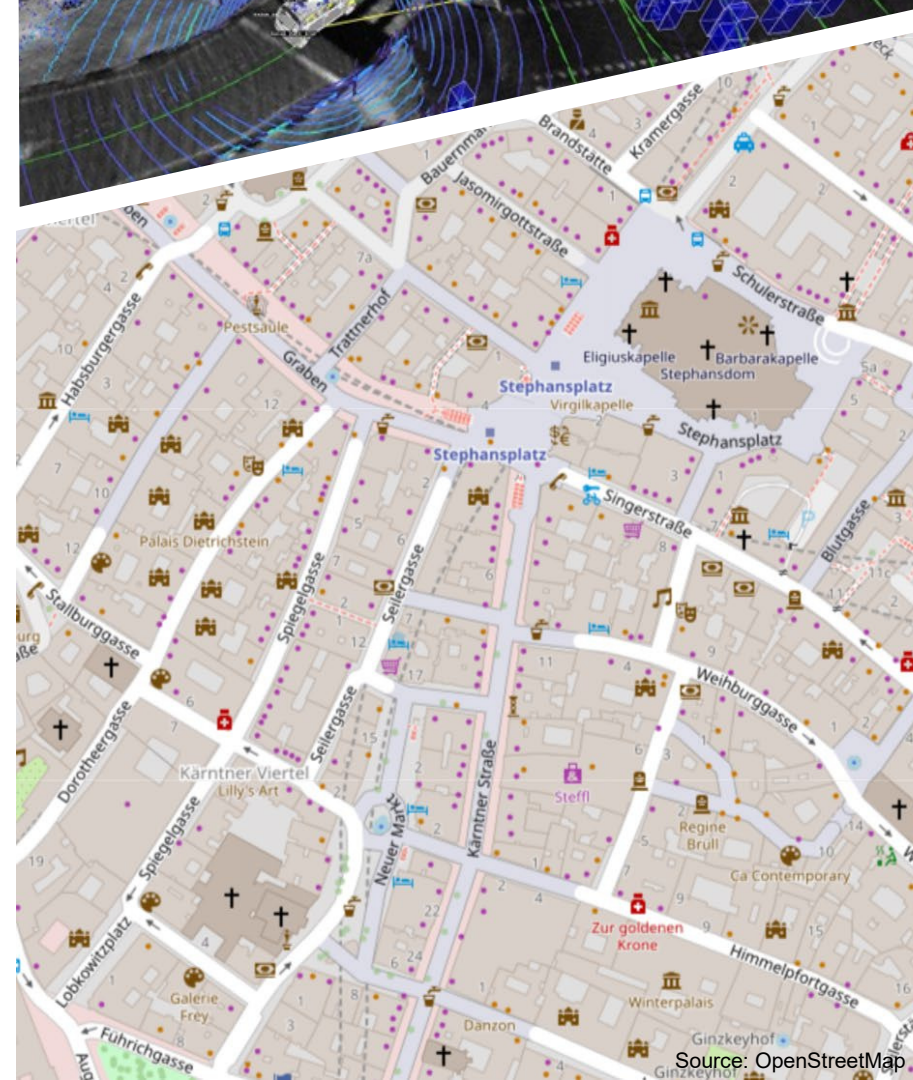
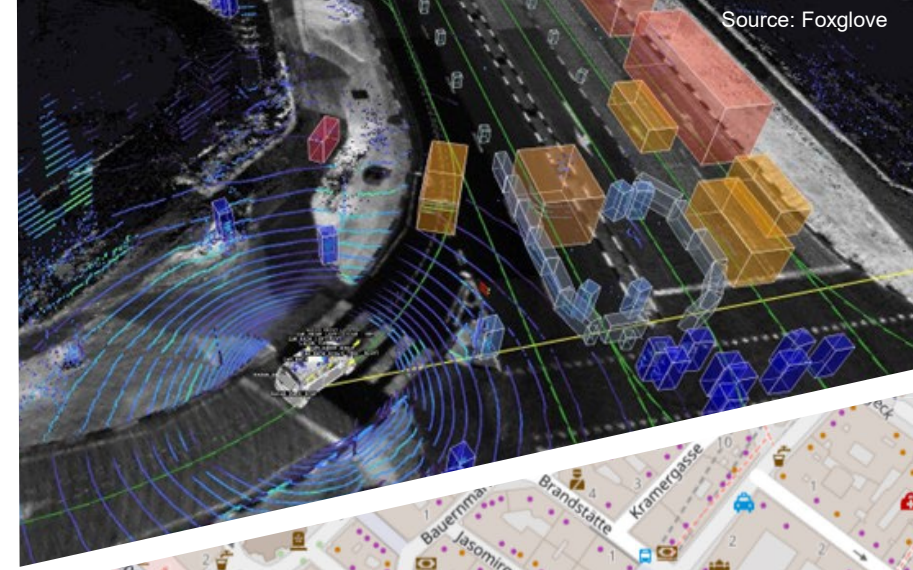
MOTIVATION – SPATIAL DATA AND ROBOTICS

- Spatial data is ubiquitous in robotics
- Temporal and short-term data is handled well by ROS2
→ Need for long-term storage, update, and distribution
- PostgreSQL DB with PostGIS extension
 - ACID compliant
 - Allows multi-user access
 - Widely used in geodesic community
 - Great tools available (e.g., QGIS)
 - Lots of open-source data available (e.g., OpenStreetMap)



OpenStreetMap

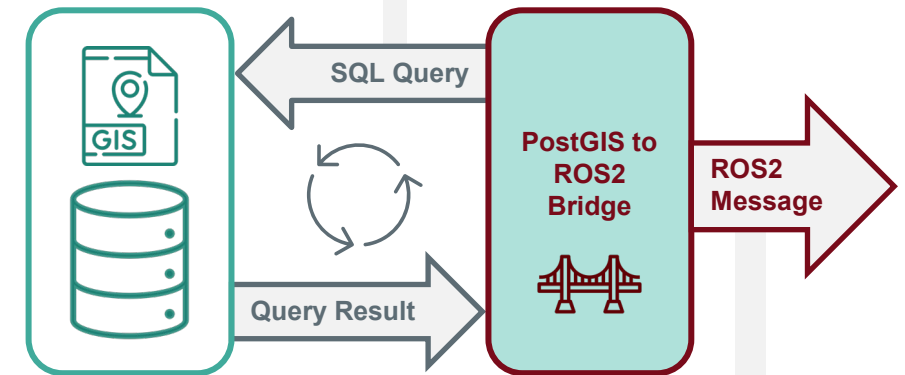
→ Need for a bridge between these ecosystems!



POSTGIS TO ROS2 BRIDGE – FEATURE OVERVIEW

- Connect to any PostgreSQL database
- Node configuration via single yaml config file
- Main interface: **SQL query**
 - Following results columns are used for
 - **geometry** 3D Position
 - **rotation** Orientation (scaled Euler rotation)
 - **frame_id** frame id of position
 - and more ROS2 messages specific data fields (optional)
- Option to set non-spatial data (e.g., **topic**) via config file
- Publish data as ROS2 messages (Point, PointArray, Marker, PointCloud, ...)

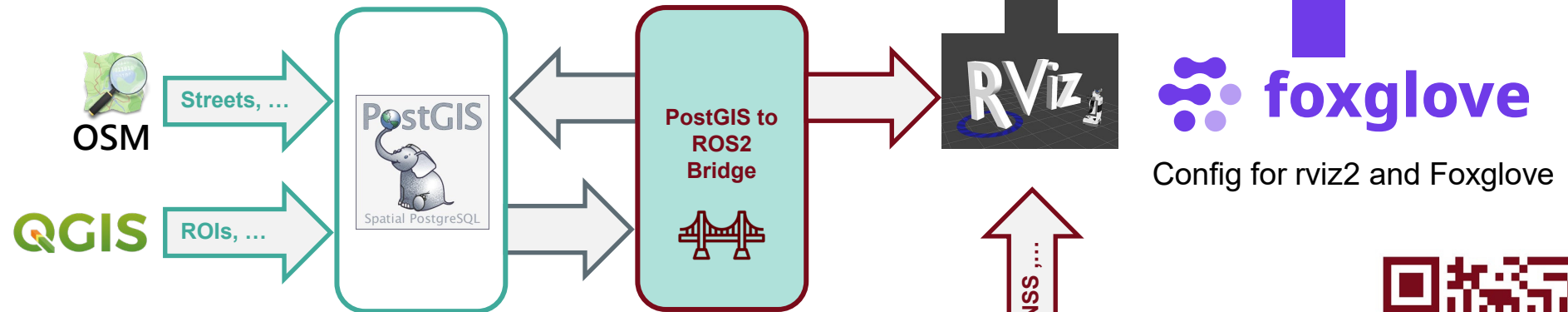
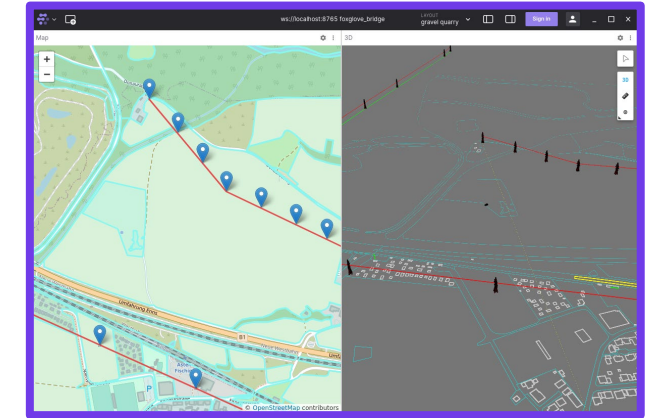
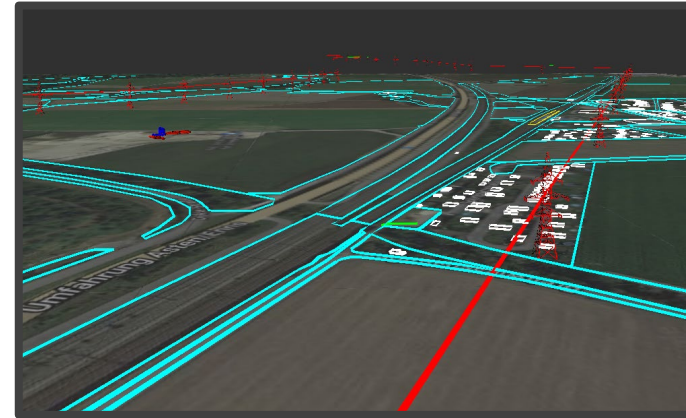
```
"SELECT position AS geometry FROM landmark;"
```



```
geometry_msgs/msg/Point  
---  
x: 184.0  
y: 456.0  
z: 3.4  
---
```


DEMO SHOWCASE “GRAVEL QUARRY”

- Simple playground environment to showcase basic functionality
https://github.com/AIT-Assistive-Autonomous-Systems/postgis_ros_bridge_demo_workspace



Spatial data in PostgreSQL DB:

- Streets, powerlines, regions, buildings from OpenStreetMap
- Customized areas defining no-go zones in gravel quarry

 **ROS 2**

“Live” data from bag file:

- GNSS (NavSatFix)
- IMU reading



DEMO SHOWCASE "GRAVEL QUARRY"

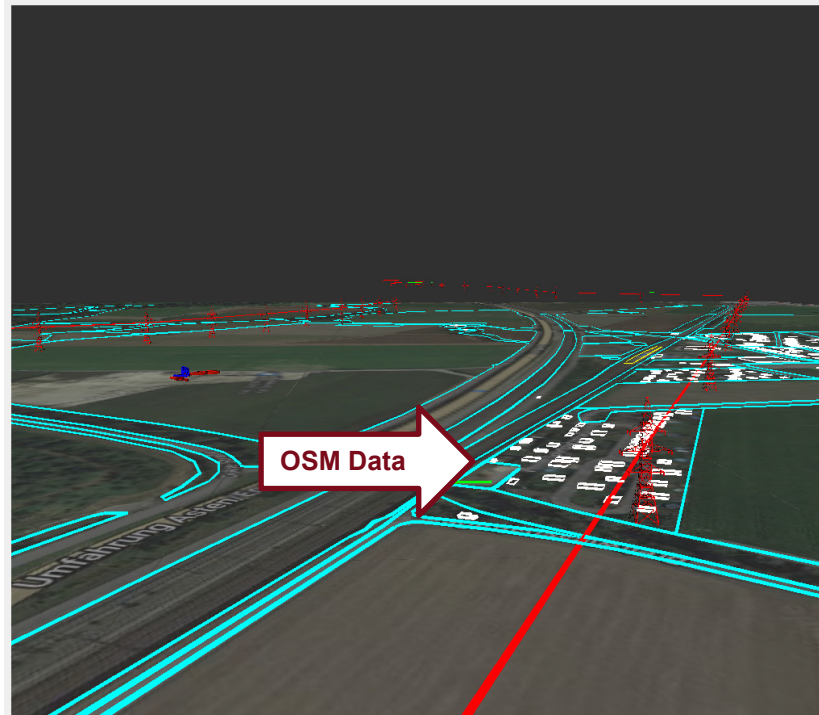


rviz2 Satellite View

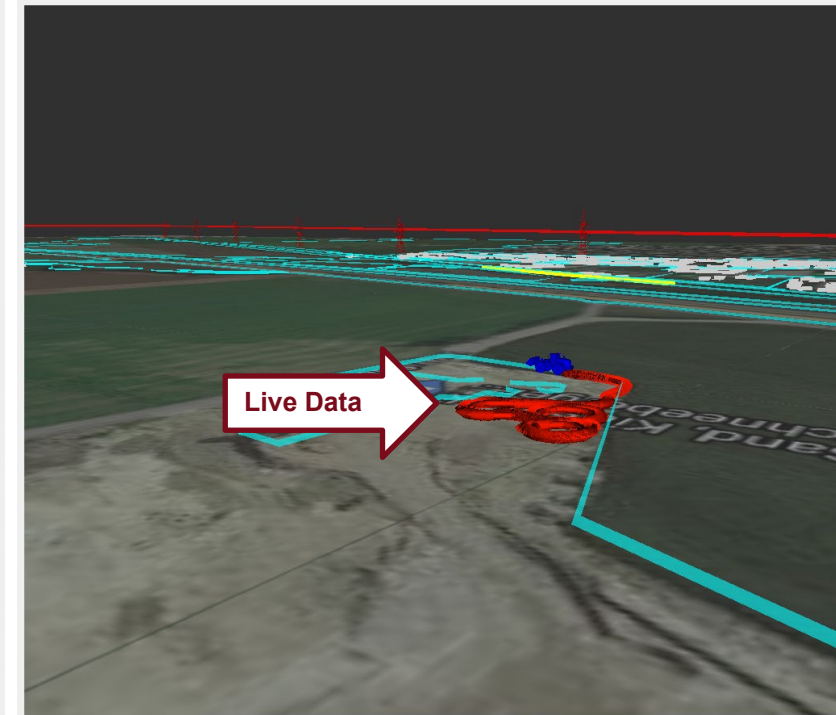


Gravel Quarry

OSM Data

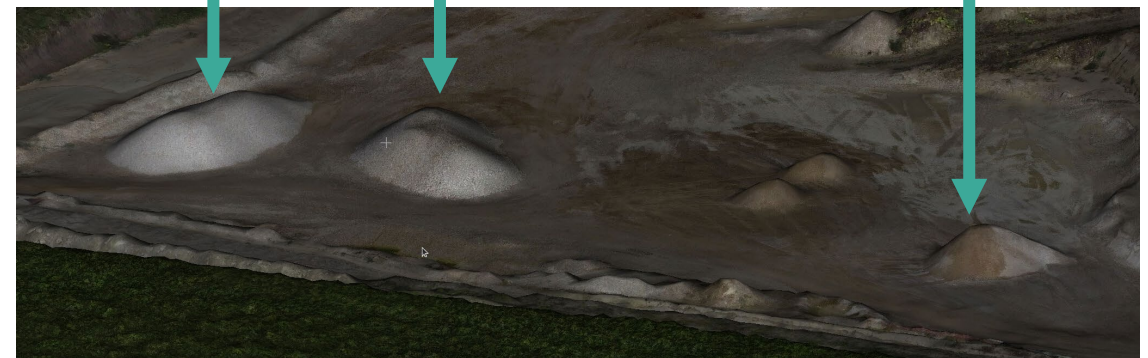
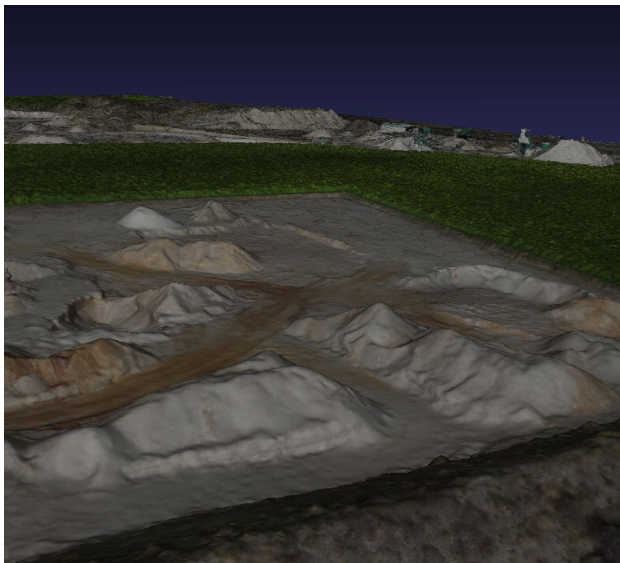


Live Data



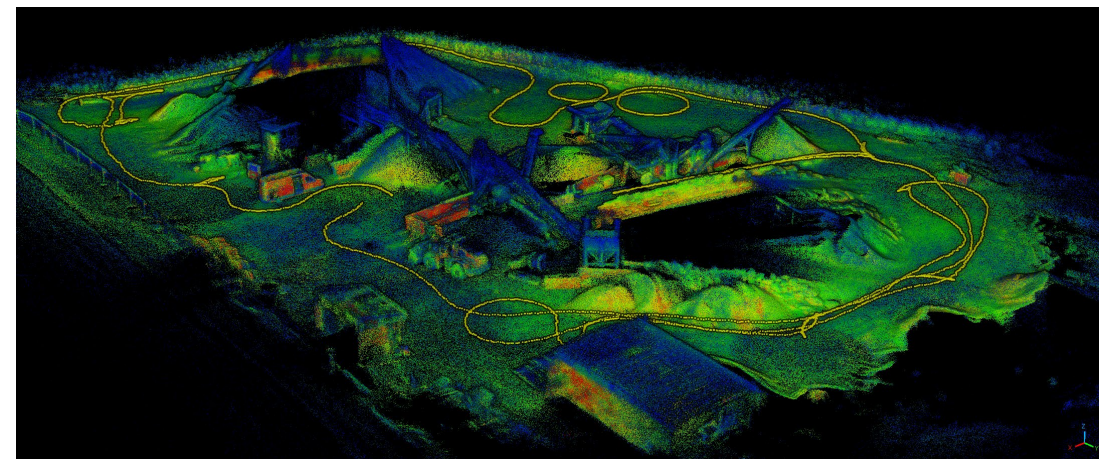
REAL WORLD USE CASES

- Definition of loading zones in PostgreSQL database
- Volume estimation and continuous stocktaking using live sensor data



REAL WORLD USE CASES

- Unified storage of geometric and semantic data in PostgreSQL/PostGIS database
 - Landmarks from visual and LiDAR SLAM
 - POIs, semantic information
 - High definition orthographic maps
 - ...



openSCHEMA



CONFIGURING THE BRIDGE

```

postgresql:
  user: "postgres"
  pass_env: "POSTGRES_PASSWORD"
  pass: "postgres"
  host: "localhost"
  port: 5432
  schema: "example_schema"

query_defaults:
  rate: 10.0
  frame_id: "map"

publish:
  - query_pose_array
  - query_polygon

query_pose_array:
  query: "SELECT pose.position AS geometry,
         pose.rotation AS rotation FROM pose;"
  type: "PoseArray"
  topic: "/pose_array"
  frame_id: "world"

query_polygon:
  ...

```

PostgreSQL specific configuration

Default values can be set for all non-spatial data

List of all publisher to be configured and set up

Configuration of a ROS2 **PoseArray** publisher:

- Query fetches position and rotation from “**pose**” table
- Result is published at topic “**pose**” with frame “**world**”

Configuration section of next publisher...

VERSATILITY AND UTM TRANSFORMER

- Data is often available in geodesic coordinates

```
cartesian_transform:
```

```
  type: "utm"
```

```
  # utm_zone: 33
```

```
  # utm_band: "N"
```

```
  lon: 16.51142
```

```
  lat: 47.97727
```

```
  broadcast_cartesian_transform: true
```

```
  yaw_offset: 0.0
```

```
  cartesian_frame_id: "utm"
```

```
  world_frame_id: "map"
```



Type of the cartesian coordinate

Directly set UTM zone and band

Used to auto detect UTM zone and band (if not set explicit)

Broadcast transformation from
cartesian_frame_id → world_frame_id

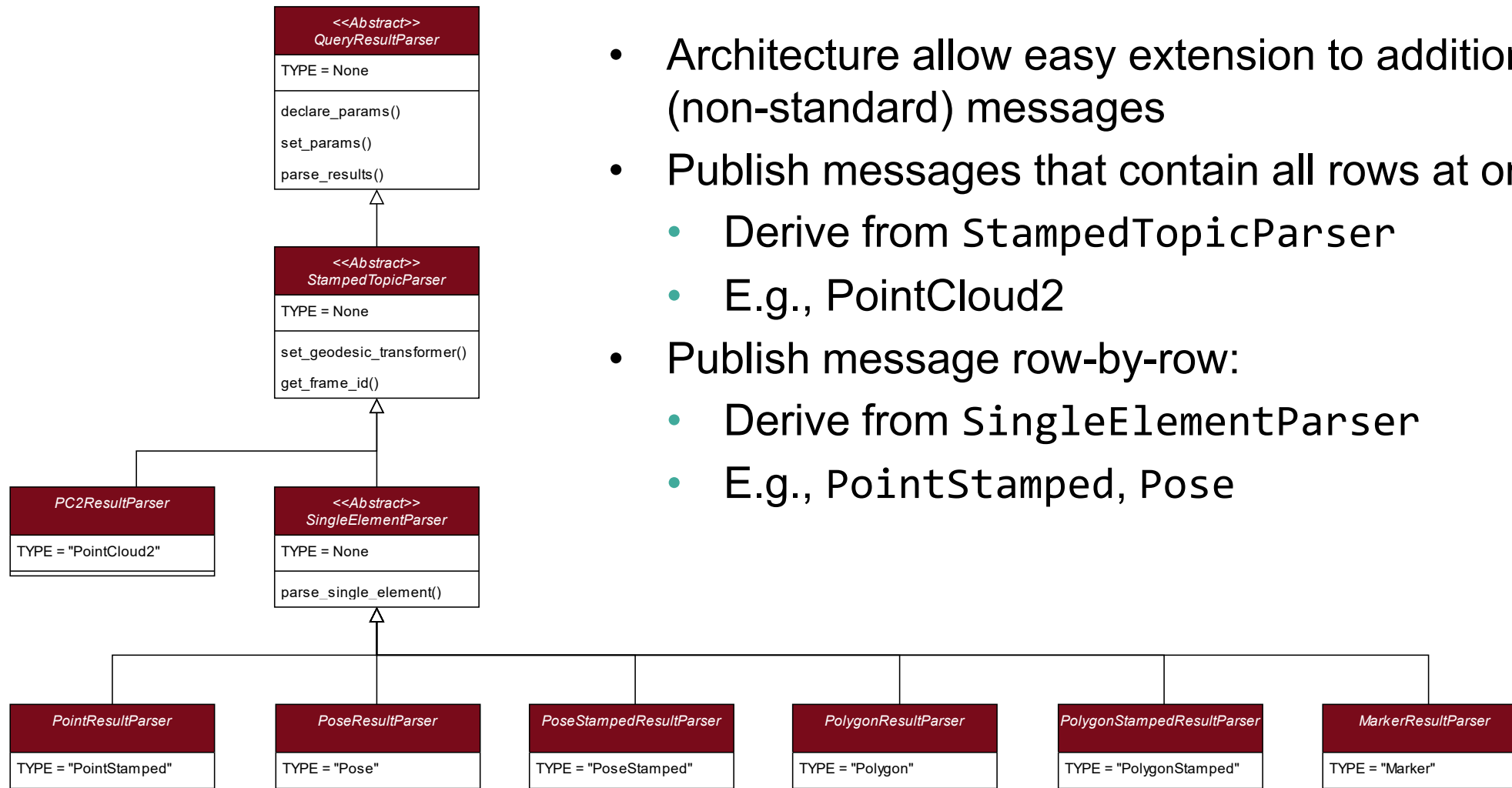
Additional rotational offset between cartesian and world frame

Name of cartesian frame and transformed frame

➔ Applied for all queries with **geodesic=True** set.

Alternative: Apply transformation using PostGIS functions in query (ST_Transform).

EXTENSIBILITY OF THE BRIDGE



- Architecture allow easy extension to additional (non-standard) messages
- Publish messages that contain all rows at once:
 - Derive from StampedTopicParser
 - E.g., PointCloud2
- Publish message row-by-row:
 - Derive from SingleElementParser
 - E.g., PointStamped, Pose

EXTENSIBILITY OF THE BRIDGE

- Add a new parser - step by step:
 1. Add new converter class to `query_result_parser.py` and derive from
 - `StampedTopicParser`, if all rows get into one message
 - `SingleElementParser`, to publish row-by-row
 2. Register the parser in the node by adding it to the `query_parser` dictionary
 3. (Optional) Generate an array-version with the `BasicStampedArrayParserFactory` (see example `Marker` → `MarkerArray`)
 4. Update the YAML file to launch the new parser
- Planned features
 - Publish on change / on demand

CONCLUSION AND Q&A

- Bridging spatial long-term data to multi-agent ROS2 live systems
- Easy to use and easy to extend software architecture
- VSCode devcontainer with demo workspace and data available:
https://github.com/AIT-Assistive-Autonomous-Systems/postgis_ros_bridge_demo_workspace
- Questions? Don't hesitate to contact us!

MARCO WALLNER

Scientist

Assistive & Autonomous Systems

Center for Vision, Automation & Control

AIT Austrian Institute of Technology GmbH

Giefinggasse 4 | 1210 Vienna | Austria

T +43 50550-4184 | M +43 664 88256055

marco.wallner@ait.ac.at | www.ait.ac.at



GitHub